

Señales y Sistemas

Laboratorio 0 (2h)

Profesor responsable: Luis Vielva

Curso 2002/2003

Materiales: (a) El alumno debe traer unos auriculares similares a los de un reproductor portatil. (b) Todos los ordenadores están equipados con Matlab y tarjeta reproductora de sonido.

Objetivos: (a) Tomar contacto con el Laboratorio del Grupo de Tratamiento Avanzado de Señal. Adquirir la dinámica de utilización de los ordenadores, organización de las cuentas de usuario, directorios y ejecución de las aplicaciones que se van a utilizar. (b) Identificar Matlab como una herramienta de cómputo y visualización en el estudio de señales y sistemas. (c) Comprobar el efecto de manipulaciones sencillas sobre señales.

Desarrollo

Utilización del ordenador: Cada alumno tiene asignado una cuenta. El nombre de usuario para la cuenta es SSXXX, donde XXX representa el número de orden en la lista de alumnos de la asignatura. Así por ejemplo, el tercer alumno de la lista utilizará como nombre de usuario SS003.

La contraseña la elige el alumno, el primer día que se conecta debe dejar en blanco la casilla en la que le pide la contraseña, y a continuación escribir la contraseña que quiere tener. Es importante recordar esta contraseña, ya que el profesor no tiene forma de acceder a ella.

Características de matlab: la mejor forma de aprender Matlab es experimentando por uno mismo. En esta pequeña guía se indican órdenes para escribir en la línea de comandos. Estas mismas órdenes también se pueden escribir en un fichero (macros y funciones) para ejecutarlas más sencillamente. Ten en cuenta los siguientes aspectos:

- Las mayúsculas y minúsculas no son equivalentes.
- Al escribir el nombre de una variable, Matlab mostrará su valor actual.
- Un punto y coma al final de una orden suprime la salida por pantalla.

- Los paréntesis y corchetes no son intercambiables.
- Las flechas arriba y abajo sirven para recorrer una lista con las órdenes previamente ejecutadas.
- Si escribes `help tema` accederás a la ayuda en línea sobre la orden, función o símbolo `tema`.

Conceptos básicos: La estructura de datos más utilizada en Matlab son las matrices (los vectores y escalares son casos particulares de matrices). Ejecuta las siguientes órdenes y observa el resultado proporcionado por Matlab:

- Matlab soporta aritmética compleja `>>w = (-1)^0.25`
- Construye un vector fila `>>a = [1 2 3]`
- Construye un vector columna `>>c = [4; 5; 6]`
- Multiplica (producto interno) ambos vectores `>>a*c`
- Calcula el producto externo `>>A=c*a`
- Observa el error que obtienes si no coinciden las dimensiones para una multiplicación matricial `>>a*a`
- Aplica una operación a cada elemento de un vector (observa el punto antes del operador elevar) `b=a.^2`
- También puedes multiplicar dos vectores elemento a elemento `a.*a`
- Muchas funciones matemáticas de matlab operan sobre vectores `exp(a)`
- Puedes operar sobre el resultado anterior `log(ans)`
- Existen funciones para analizar datos `sum(b)`, `mean(c)`
- Construye una matriz `>>B=[-3 0 1; 2 5 -7; -1 4 8]` y resuelve el sistema lineal $Bx = c$ mediante `>>x=B\c`
- Calcula los autovalores de B `>>e = eig(B)`
- El operador dos puntos es útil para construir vectores de valores equiespaciados `>>v=1:6`
- Puedes conseguir cualquier incremento `>>w=2:3:10`, `>>y=1:-0.25:0`
- Puedes construir matrices mayores a partir de otras más pequeñas `>>C=[A, [8; 9; 10]]`, `>>D=[B;a]`

- Los elementos de una matriz se seleccionan mediante índices, `>>C(2,3)`
- En general, `C(i1:i2, j1:j2)` selecciona la submatriz formada por la intersección de las columnas desde `i1` hasta `i2` y las columnas desde `j1` hasta `j2`. Prueba con `>>C(2:3, 1:2)`
- Utilizar el operador `:` en una coordenada selecciona todos los valores `>>C(1,:)`, `>>C(2:3,:)`, `>>C(:, [1,4])`
- Algunas matrices tienen nombres predefinidos, `>>I3=eye(3, 3)`, `>>Y=zeros(3, 5)`, `>>Z=ones(2)`
- Puedes generar números aleatorios distribuidos uniformemente en el rango `[0,1]`, `>>rand(3)`, o con distribución normal de media cero y varianza unidad `>>randn(1, 6)`
- Puedes ver qué variables tienes definidas con `>>who`
- Matlab tiene lazos, que deberías tratar de utilizar lo menos posible, ya que penalizan notablemente la eficiencia

```
ejex = [];
j = 1;
for t=0:0.005:1
    ejex = [ejex, t];
    z(j) = exp(10*t*(t-1)) * sin(12*pi*t);
    j = j + 1;
end
plot(ejex, z)
```

- Prueba `t=0:0.005:1;z=exp(10*t.*(t-1)).*sin(12*pi*t);plot(t,z,'r')`

Manipulación de señales:

- Puedes mostrar un fichero gráfico pregrabado, como el mapa de la radiación del fondo cósmico de microondas

```
>>close all;
>>image(imread('cmbr.bmp'));
>>xlabel('x'); ylabel('y');
```

- Lee un fichero de sonido del archivo `anykey.wav`, los parámetros devueltos son el vector con las muestras y la frecuencia de muestreo (cuyo inverso es el intervalo de muestreo)

```

close all;
[y,fs] = wavread('anykey');
n = length(y)
fs
dt = 1 / fs;
plot(dt*[1:n], y); axis tight;

```

- También puedes escuchar el sonido >>soundsc(y, fs)
- O dibujar la señal en una ventana con dos gráficas: la forma de onda y el espectrograma

```

close all;
subplot(2, 1, 1);
plot(dt*[1:n], y); axis tight;
xlabel('Tiempo (s)'); ylabel('Amplitud');
subplot(2, 1, 2); specgram(y);

```

- Lee un fichero con las cotizaciones de bolsa de microsoft y representa su evolución en el tiempo, tanto en escala lineal como logarítmica

```

close all;
fid = fopen('msft.clo', 'r');
x = fscanf(fid, '%f', inf);
fclose(fid);
x = x(end:-1:1);
subplot(2, 1, 1); plot(x);
subplot(2, 1, 2); semilogy(x);

```

- Matlab incorpora herramientas muy potentes de procesamiento de señal, que no utilizaremos de momento, tal como un detector de bordes

```

close all;
x = imread('rice.tif');
figure(1); subplot(1, 2, 1); imshow(x);
drawnow; y = edge(x, 'canny');
subplot(1, 2, 2); imshow(y);

```

- Modelamos un eco en Matlab (no hace falta que lo teclees, está escrito en el fichero eco.m, por lo que es suficiente con que escribas >>eco)

```

close all;
[y, fs] = wavread('carta');

```

```

y1 = y(1:end-3000);
y2 = y(3001:end);
z = y1 + y2;
subplot(3, 1, 1); plot(y1, 'b'); pause;
soundsc(y1, fs); pause;
subplot(3, 1, 2); plot(y2, 'g'); pause;
subplot(3, 1, 3); plot(z, 'r'); pause;
soundsc(z, fs);

```

- Observa el proceso de muestrear una señal continua (fichero `sampling.m`)

```

close all;
ezplot('exp(-t*t/20)*sin(1.5*t)', [-10 10]);
set(gca, 'xgrid', 'on');
hold on;
n=-10:10;
stem(n, exp(-n.*n/20).*sin(1.5*n), 'r', 'filled');
xlabel('t ó n');

```

- Observa el proceso de cuantizar y discretizar una señal analógica mediante la operación $x[n] = \Delta \text{round}(x(nT)/\Delta)$ (fichero `cuantos.m`)

```

close all;
ezplot('5*exp(-t*t/20)*sin(1.5*t)', [-10 10]);
set(gca, 'ygrid', 'on');
hold on;
n=-10:10;
y = 5*exp(-n.*n/20).*sin(1.5*n);
stem(n, y, 'r');
stem(n, quant(y, 1), 'g');

```

- Observa los cuatro tipos de señales (fichero `contdisc.m`)

```

close all;

t = linspace(0, 2*pi, 200);
x = sin(t);
T = pi/8;
D = 0.2;
subplot(2, 2, 1); plot(t, x);
xlabel('t'); title('Analógica'); axis tight;
subplot(2, 2, 2); plot(t, D * round(x / D));
xlabel('t'); title('Cuantizada'); axis tight;

```

```

n = 0:floor(2*pi/T);
y = sin(n*T);
subplot(2, 2, 3); plot(t/T, x, ':'); hold on;
stem(n, y, 'r'); axis tight;
xlabel('n'); title('Muestreada');
subplot(2, 2, 4); plot(t/T, D*round(x/D), ':'); hold on;
stem(n, y, 'r'); axis tight;
xlabel('n'); title('Digital');

```

- Observa los desplazamientos temporales continuos (fichero `shiftc.m`) y discretos (fichero `shiftd.m`).
- Observa una inversión temporal

```

close all;
[y, fs] = wavread('carta');
subplot(2, 1, 1); plot(y); pause;
soundsc(y, fs); pause;
z = y(end:-1:1);
subplot(2, 1, 2); plot(z); pause;
soundsc(z, fs);

```

- Observa el escalado temporal en el fichero `escala.m`
- Observa su efecto sobre una señal de audio

```

close all;
[y,fs] = wavread('carta');
y = y(1:27700);
n = 1:length(y);
subplot(3, 1, 1); plot(n/fs, y); soundsc(y, fs); axis tight; pause;
subplot(3, 1, 2); plot(n/(1.5*fs), y); soundsc(y, 1.5*fs); axis tight; pause;
subplot(3, 1, 3); plot(n/(0.7*fs), y); soundsc(y, 0.7*fs); axis tight;

```